

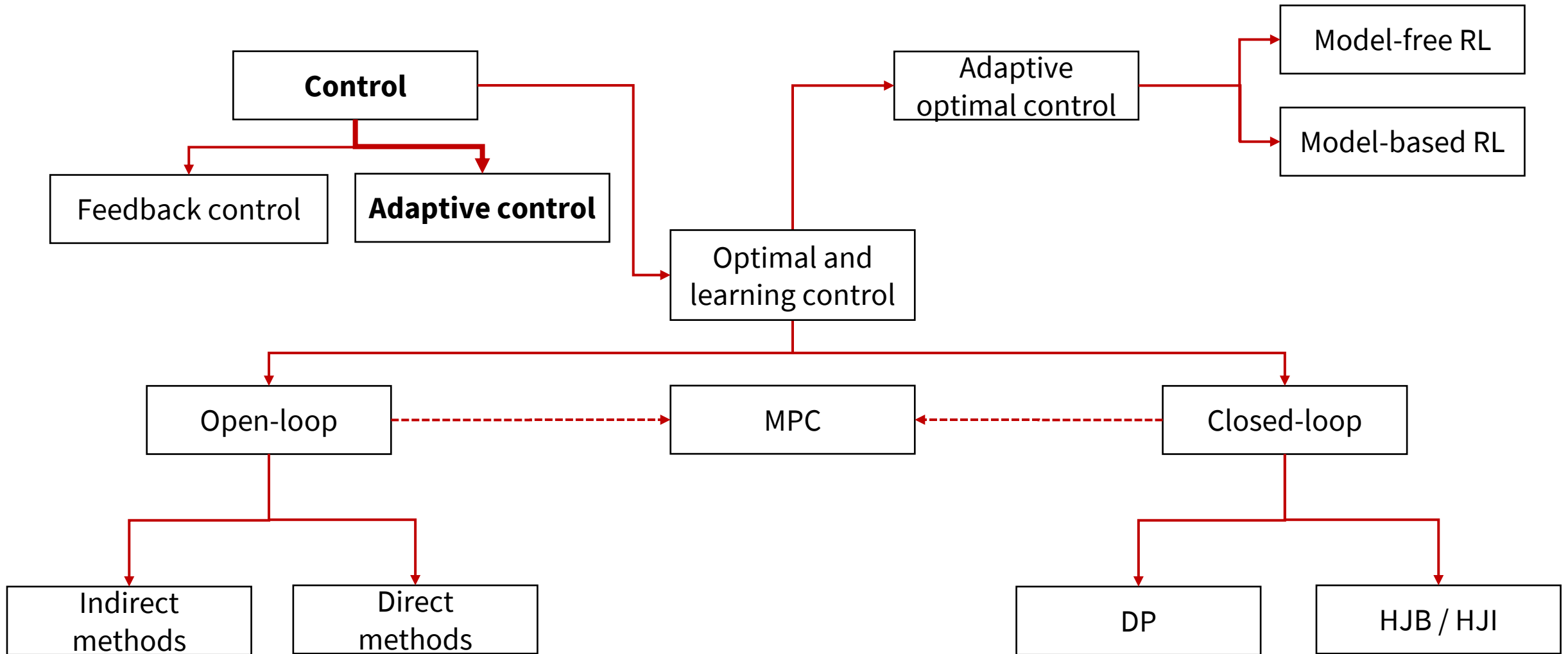
# AA203

# Optimal and Learning-based Control

Intro to learning; System identification and adaptive control

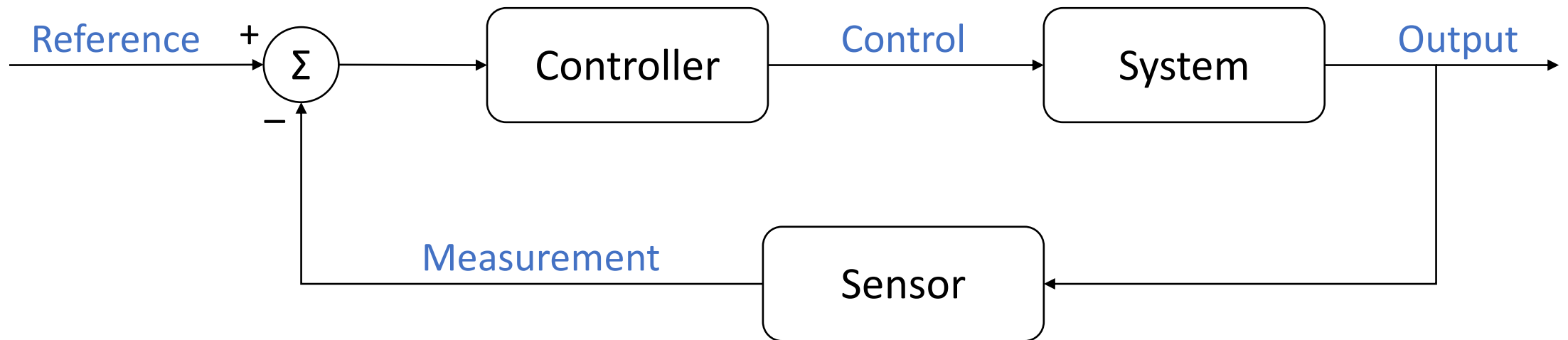


# Course overview



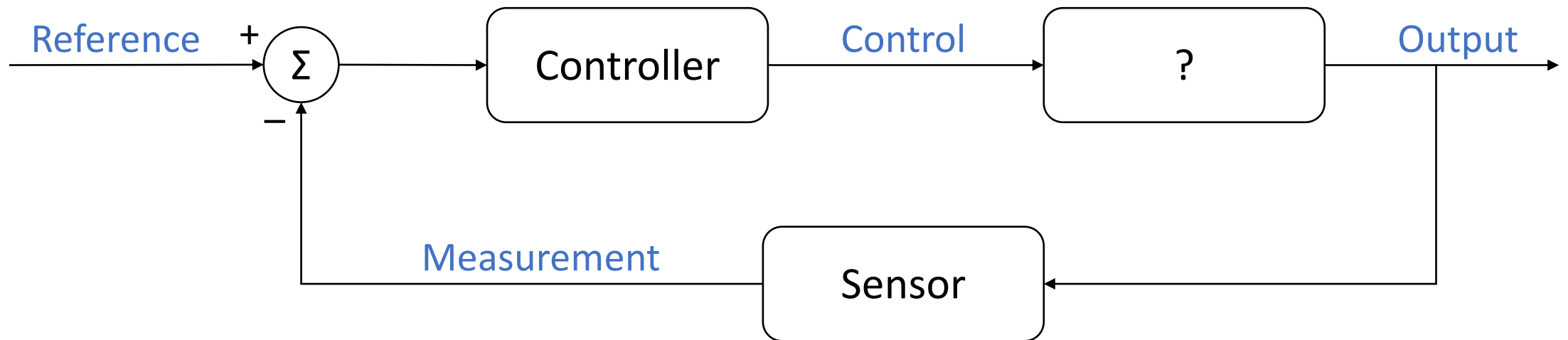
# Feedback control

- Reference tracking, with uncertainty

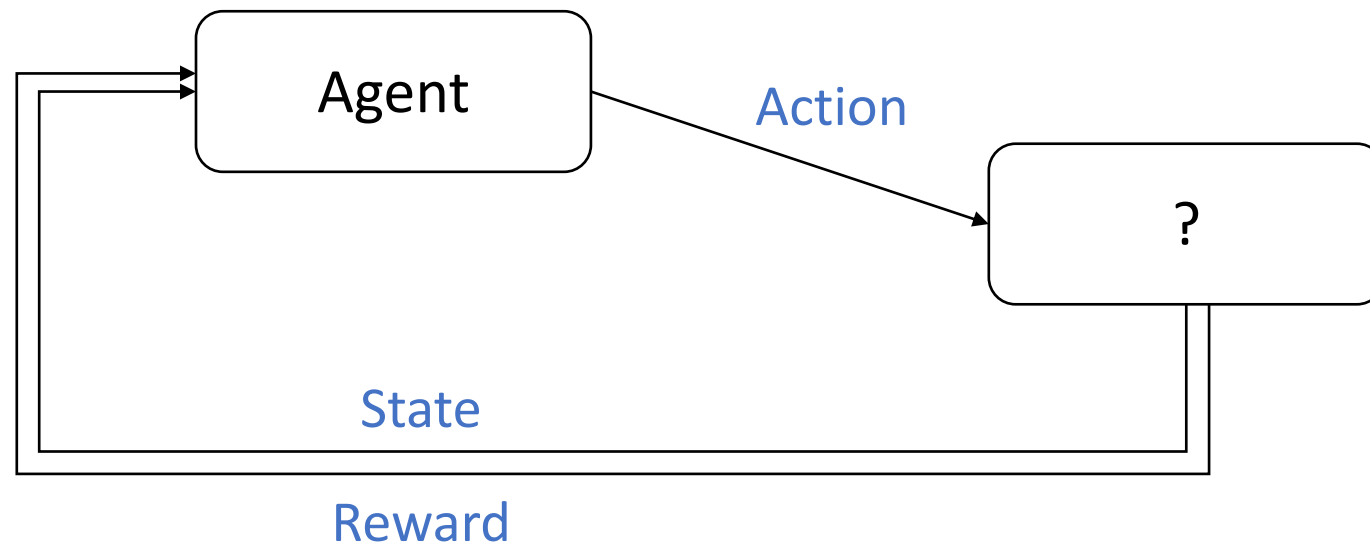


# Feedback control

- Reference tracking, with uncertainty



# Reinforcement learning



# Approaches

How do we handle uncertainty?

- In many cases, when uncertainties have only a small effect, a feedback controller will adequately compensate for model error
  - Small wind disturbances in a quadrotor
- We can use robust control approaches (e.g., minimax control strategies)
  - Harrier VSTOL near hover
- We can use observed state transitions to attempt to identify patterns and improve our control strategy
  - F-16 under wind resistance



# What can we learn?

Want to use measurements to improve control performance.

Can either:

- Use measurements to directly improve controller
  - Direct adaptive control
  - Model-free reinforcement learning
- Use measurements to learn model, use model to improve controller
  - System identification
  - Indirect adaptive control
  - Model-based reinforcement learning

# How does learning happen?

Three possible learning settings:

- **“Zero” episodes:** the **system identification** approach, in which learning is done based on data gathered before operation
- **One episode:** want to learn and re-optimize our controller online -> this is the standard setting for **adaptive control**
- **Multiple episodes:** interact with the environment in episodes, in which the system is reset at the start of each episode; learning and policy optimization can happen between episodes -> this is the standard setting for **reinforcement learning**



# System identification for learning-based control

- For many problems, we don't need to learn online
- A standard control engineering pipeline is to do experiments in advance to build a *data-driven* model of the dynamics
- Then, we can use this model for planning and control without further learning
- Relies on having an engineer in the loop for learning, designing experiments, resetting the system, etc.
- *Linear regression* is one of the main system id tools

# Least squares

System model

$$y = \boldsymbol{\theta}^T \mathbf{z} + \epsilon$$

Given data  $y_1, \dots, y_N, \mathbf{z}_1, \dots, \mathbf{z}_N$ , want to minimize mean squared error:

$$\sum_{i=1}^N (y_i - \boldsymbol{\theta}^T \mathbf{z}_i)^2$$

Rewrite as

$$\|\mathbf{y} - \mathbf{Z} \boldsymbol{\theta}\|_2^2$$

Solution (full rank  $X$ ):  $\hat{\boldsymbol{\theta}} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{y}$

# Example: first order model

- Consider system with dynamics

$$x(t + 1) = ax(t) + bu(t) + v(t)$$

- Linear regression representation

- $\mathbf{z}_t = [x(t), u(t)]$ ,  $t = 0, \dots, N$

- $\boldsymbol{\theta} = [a, b]'$

- Practically, least squares can be written in recursive form for efficiency

# Discrete time from continuous time

# Linear regression for system id

- As seen before, the solution is

$$\hat{\boldsymbol{\theta}} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{y}$$

- Gauss-Markov theorem:  $\hat{\boldsymbol{\theta}}$  is the **best linear unbiased estimator** (for any noise distribution that obeys assumptions)
- If noise distribution is Gaussian,  $\hat{\boldsymbol{\theta}}$  is the maximum likelihood estimator

# Persistent excitation

# Persistent excitation

# Performance questions

The system identification approach leads to several questions:

- How much data is required to learn the model? How can we quantify a “good” estimate? We care about controller performance, not model accuracy, so do we require an accurate model?
- How should we design the inputs used for data collection? What if an engineer can’t intervene to prevent system failure during data collection?
- What if our system does not fall in the class of systems we are considering?



# Detour: stability analysis via Lyapunov

Mass/Spring/Damper

# Lyapunov theorem for global stability

Equilibrium point:  $\mathbf{x}^*$  s.t.  $f(\mathbf{x}^*) = 0$

1.  $V(\mathbf{x}) > 0$
2.  $\dot{V}(\mathbf{x}) < 0$
3.  $V(\mathbf{x}) \rightarrow \infty$  as  $\|\mathbf{x}\| \rightarrow \infty$

If such a function exists, then equilibrium  $\mathbf{x}^* = 0$  is globally asymptotically stable.

# Example: 1-d double integrator

# Adaptive control

- Broadly, adaptive control aims to perform *online adaptation* of the policy to improve performance
- This can be done via directly updating the policy or updating the model and re-optimizing or re-computing the controller
- Most adaptive control work does not consider the *optimal adaptive control* problem; they focus on proving stability of the coupled controller and adaptive component

# Adaptive control approaches

Encompasses a large variety of techniques, including :

- Model reference adaptive control (MRAC)
- Model identification adaptive control (MIAC)
- Dual control
- Model-free
  - policy adaptation
  - iterative learning control

# Model reference adaptive control (MRAC)

- A model reference adaptive controller is composed of four parts:
  1. A plant containing unknown parameters
  2. A *reference model* for compactly specifying the desired output
  3. A feedback control law containing adjustable parameters
  4. An adaptation mechanism for updating the adjustable parameters
- The reference model provides the ideal plant response which the adaptation mechanism should seek in adjusting the parameters

# Example of MRAC control

- Consider double integrator

$$m\ddot{x} = u$$

- Assume a human operator provides the positioning command  $r(t)$  to the control system
- A reasonable way of specifying the ideal response of the controlled mass to the external command  $r(t)$  is to use the reference model

$$\ddot{x}_m + k_d \dot{x}_m + k_p x_m = k r(t)$$

where the reference model output  $x_m$  is the *ideal* output

# Example of MRAC control

- If the mass is known exactly, one can achieve perfect tracking via

$$u = m(\ddot{x}_m - 2\lambda\dot{\tilde{x}} - \lambda^2\tilde{x})$$

where  $\lambda > 0$  and  $\tilde{x} := x - x_m$  is the tracking error

- This control leads to exponentially convergent tracking dynamics

$$\ddot{\tilde{x}} + 2\lambda\dot{\tilde{x}} + \lambda^2\tilde{x} = 0$$



# Example of MRAC control

- If the mass is *not* known exactly, we can use the control law

$$u = \hat{m}(\ddot{x}_m - 2\lambda\dot{\tilde{x}} - \lambda^2\tilde{x})$$

which contains the adjustable parameter  $\hat{m}$

- This control leads to the closed-loop dynamics

$$m\dot{s} + \lambda ms = \tilde{m}v$$

where:

- $s$  is a combined tracking error measure, defined by  $s = \dot{\tilde{x}} + \lambda\tilde{x}$
- the signal quantity  $v$  is given by  $v = \ddot{x}_m - 2\lambda\dot{\tilde{x}} - \lambda^2\tilde{x}$
- and the parameter estimation error is  $\tilde{m} = \hat{m} - m$

# Example of MRAC control

- One way of adjusting parameter  $\hat{m}$  is to use the (nonlinear) update law

$$\dot{\hat{m}} = -\gamma v s$$

where  $\gamma > 0$  is called the *adaptation gain*

- Stability and convergence can be analyzed via Lyapunov theory
- Consider Lyapunov function candidate

$$V = \frac{1}{2} \left[ m s^2 + \frac{1}{\gamma} \tilde{m}^2 \right]$$

- Its derivative is  $\dot{V} = -\lambda m s^2$
- Thus  $s \rightarrow 0$ , and hence  $\tilde{x} \rightarrow 0$  and  $\dot{\tilde{x}} \rightarrow 0$

# MRAC

- An excellent reference for systematic MRAC design is: Jean-Jacques Slotine, Weiping Li, Applied Nonlinear Control, Chapter 8
- If the reference signal  $r(t)$  is very simple, such as zero or a constant, it is possible for many vectors of parameters, besides the ideal parameter vector, to lead to tracking error convergence
- However, if the reference signal  $r(t)$  is so complex that only the “true” parameter vector can lead to tracking convergence, then one shall have parameter convergence -> *persistent excitation condition*

# Model identification adaptive control

- MIAC (also referred to as self-tuning) simply combines model estimation with a controller that uses the estimated model
- Important distinction between *certainty-equivalent* and *cautious* approaches
  - **Certainty-equivalent:** maintains point estimate of model and uses that model for policy selection/optimization. Note that unlike the LQG setting, certainty-equivalence is sub-optimal.
  - **Cautious:** Maintains measure of estimator uncertainty, incorporates the uncertainty into the controller. This is often overly robust because it does not account for future info gain!

# MRAC vs. MIAC

- MRAC and MIAC arise from two different perspectives:
  1. parameters in MRAC are updated so as to minimize tracking error between the plant output and the reference model output
  2. parameters in MIAC are updated so as to minimize the data-fitting error
- MIAC controllers are in general more flexible, as one can couple various controllers with various estimators
- However, correctness of MIAC controllers is more difficult to guarantee, as if the signals are not rich, the estimated parameters may not be close to the “true” values, and stability and convergence may not be ensured
- In contrast, for MRAC, stability and convergence are usually guaranteed *regardless* of the richness of the signals

# Next time

- Optimization